IN THE CLAIMS:

Please AMEND the claims as indicated below:

1. (currently amended) A file system using a computer readable storage medium, the file system which operates on an operating system, comprising:

an access executing unit which, when an access to a file occurs, processes said file in accordance with said access;

a user defined process holding unit which holds a user defined process which has previously been defined by the user;

a file managing unit which manages said file with two areas including said data area and a meta data area, and provides an extended meta data area in said meta data area to store extended meta data correlating the file managed in said data area and said user defined process held in said defined process holding unit; and

a defined process executing unit which executes said user defined process correlated by the extended meta data of said file managing unit by using the access to said file as a trigger.

2. (previously presented) A system according to claim 1, wherein said file managing unit enables the user to designate a format of said extended meta data area.

3. (original) A system according to claim 2, wherein said file managing unit designates the format of said extended meta data area in accordance with contents in said data area.

4. (original) A system according to claim 3, wherein said file managing unit sets meta data, namely, a file type as a format of said extended meta data area and determines the format of said extended meta data area in accordance with said file type.

5. (original) A system according to claim 4, wherein said file managing unit determines the file type upon creation of the file, sets the extended meta data area in accordance with said file type, and thereafter, enables the user to change said file type and change said extended meta data area.

6. (original) A system according to claim 5, wherein said file managing unit automatically determines the file type from the contents in the data area of said file and also automatically

determines said extended meta data area.

7. (original) A system according to claim 5, wherein said file managing unit has a tree structure using a directory for managing a plurality of files, and the file type which is set upon creation of the file succeeds a file type of a parent directory.

8. (original) A system according to claim 2, wherein said file managing unit holds information extracted from said data area as extended meta data into said extended meta data area.

9. (original) A system according to claim 8, wherein said extended meta data extracted from said data area is duplicated and held in said data area and said extended meta data area.

10. (original) A system according to claim 8, wherein in said extended meta data extracted from said data area, a substance is held in said data area and a pointer to the substance in said data area is held in said extended meta data area.

11. (original) A system according to claim 2, wherein upon writing into the data area of said file, said defined process executing unit changes extended meta data in said meta data area on the basis of said user defined process.

12. (original) A system according to claim 2, wherein upon writing into the data area of said file, said defined process executing unit sends a message to a user program which is additionally provided and changes extended meta data in said extended meta data area.

13. (original) A system according to claim 2, wherein upon writing into the data area of said file, said defined process executing unit sets a data area change flag to a high level and changes extended meta data extended in said extended meta data area by using the fact, as a trigger, that said flag has been set to the high level by a user program which is additionally provided.

14. (original) A system according to claim 4, wherein said defined process executing unit executes the user defined process in accordance with said file type.

15. (original) A system according to claim 1, further having an API for allowing the user to define a process.

16. (original) A system according to claim 15, wherein said API has a double layer structure comprising an API which is executed in a kernel area and an API which is executed in a user area.

17. (original) A system according to claim 2, wherein a size of said extended meta data area is variable in accordance with the file contents.

18. (currently amended) A control method for a file system which operates on an operating system, comprising:

a user defined process holding step wherein a user defined process which has previously been defined by the user is held;

an access executing step wherein, when an access to a file occurs, said file is processed in accordance with said access;

a file managing step which manages said file with two areas including said data area and a meta data area, and provides an extended meta data area in said meta data area to store extended meta data correlating the file managed in said data area and said user defined process held in said defined process holding unit; and

a defined process executing step wherein said user defined process is executed correlated by the extended meta data of said file ~~management unit~~managing step by using the access to said file as a trigger.

19. (previously presented) A method according to claim 18, wherein said file managing step enables the user to designate a format of said extended meta data area.

20. (original) A method according to claim 19, wherein in said file managing step, the format of said extended meta data area is designated in accordance with contents in said data area.

21. (original) A method according to claim 20, wherein in said file managing step, meta data, namely, a file type is set as a format of said extended meta data area and the format of said extended meta data area is determined in accordance with said file type.

22. (original) A method according to claim 21, wherein in said file managing step, the file type is determined upon creation of the file, the extended meta data area according to said file type is set, and thereafter, the user is enabled to change said file type and change said extended meta data area.

23. (original) A method according to claim 22, wherein in said file managing step, the file type is automatically determined from the contents in the data area of said file and said extended meta data area is also automatically determined

24. (original) A method according to claim 22, wherein in said file managing step, a tree structure using a directory for managing a plurality of files is provided, and the file type which is set upon creation of the file succeeds a file type of a parent directory.

25. (original) A method according to claim 19, wherein in said file managing step, information extracted from said data area is held as extended meta data into said extended meta data area.

26. (original) A method according to claim 25, wherein said extended meta data extracted from said data area is duplicated and held in said data area and said extended meta data area.

27. (original) A method according to claim 25, wherein in said extended meta data extracted from said data area, a substance is held in said data area and a pointer to the substance in said data area is held in said extended meta data area.

28. (original) A method according to claim 19, wherein in said defined process executing step, upon writing into the data area of said file, extended meta data in said meta data area is changed on the basis of said user defined process.

29. (original) A method according to claim 19, wherein in said defined process executing step, upon writing into the data area of said file, a message is sent to a user program which is additionally provided and extended meta data in said extended meta data area is changed.

30. (original) A method according to claim 19, wherein in said defined process executing step, upon writing into the data area of said file, a data area change flag is set to a high level and extended meta data in said extended meta data area is changed by using the fact, as a trigger, that said flag has been set to the high level by a user program which is additionally provided.

31. (original) A method according to claim 21, wherein in said defined process executing step, the user defined process is executed in accordance with said file type.

32. (original) A method according to claim 19, wherein an API for allowing the user to define a process is further provided.

33. (original) A method according to claim 32, wherein said API has a double layer structure comprising an API which is executed in a kernel area and an API which is executed in a user area.

34. (original) A method according to claim 19, wherein a size of said extended meta data area is variable in accordance with the file contents.

35. (currently amended) A computer readable storage medium storing a program for realizing a file system, wherein said program allows a computer to execute:

a user defined process holding step wherein a user defined process which has previously been defined by the user is held;

an access executing step wherein, when an access to a file occurs, said file is processed in accordance with said access;

a file managing step which manages said file with two areas including said data area and a meta data area, and provides an extended meta data area in said meta data area to store extended meta data correlating the file managed in said data area and said user defined process held in said defined process holding unit; and

a defined process executing step wherein said user defined process is executed correlated by the extended meta data of said file management unit by using the access to said file as a trigger.

36. (cancelled)

37. (cancelled)

38. (cancelled)